REMARKS

This is in response to the Office Action dated August 9, 2005, claims 1, 3, 5-31, and 33, 35-41 are pending. The Examiner's reconsideration of the rejections is respectfully requested in view of the amendments and remarks.

Applicants appreciate the courtesies extended by the Examiner during the telephone conversation of November 9, 2005, during which the Examiner confirmed that prosecution of the present application has been reopened in view of the Appeal Brief filed May 19, 2005.

Claims 1, 3-5, 10, 11, 31 and 33-36 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Katzman in view of Morris et al., (U.S. Patent No. 6,286,095) and Hamacher et al., Computer Organization, 1978, McGraw-Hill, Inc., Second Edition, pages 112-114. The Examiner stated essentially that the combined teachings of Katzman, Morris and Hamacher teach or suggest all the limitations of claims 1, 3-5, 10, 11, 31 and 33-36.

Claims 1 and 31 claim, *inter alia*, "replacing the instruction referencing the stack with a references to a processor-internal registers of the first processor and entering the reference to the processor-internal register in a dispatch table upon determining that the instruction uses an architecturally defined stack access method; and performing a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, and entering the in-order value in the dispatch table upon determining that the stack reference references a register of a second processor."

Katzman teaches a method for managing a system consisting of four top-of-stack registers using a namer register, and a register storing a number of top-of-stack registers containing valid information (SR) and a register containing a top piece of stack information (SM) (see col. 1, line 61

to col. 2, line 6). Katzman does not teach or suggest "performing a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, and entering the in-order value in the dispatch table upon determining that the stack reference references a register of a second processor" as claimed in claims 1 and 31. Katzman fails to teach or suggest a system including a plurality of processors. Therefore, Katzman fails to teach or suggest all the limitations of claims 1 and 31.

Morris teaches memory barrier synchronization using forced load and store operations (see col. 5, lines 21-24). The forced load and store operations prevent out-of-order processing, thus avoiding mis-synchronization of CPUs (see col. 6, lines 58-61). Morris does not teach or suggest "performing a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, and entering the in-order value in the dispatch table upon determining that the stack reference references a register of a second processor" as claimed in claims 1 and 31. Morris teaches that the load and store operations can be executed without synchronization problems between CPUs (see col. 7, lines 5-10). Morris teaches preventing mis-synchronization between CPUs through the use of blocking, suspending all subsequent load operations until a prior operation is completed (see col. 5, lines 46-48). Morris' method of avoiding mis-synchronization of CPUs through blocking is not analogous to a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, essentially as claimed in Claims 1 and 31. Morris suspends operations to prevent CPU mis-synchronization. Through the prevention of mis-synchronization, Morris effectively preempts any need for "a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory" as claimed in claims 1 and 31. Therefore, Morris fails to cure the deficiencies of Katzman.

Hamacher teaches a CPU including general-purpose registers (see page 112, section 3.6.3, lines 4-5). Hamacher does not teach or suggest "performing a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, and entering the in-order value in the dispatch table upon determining that the stack reference references a register of a second processor" as claimed in claims 1 and 31. Similar to Katzman, Hamacher fails to teach or suggest a system comprising a plurality of processors. Thus, Hamacher does not teach or suggest, "performing a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, and entering the in-order value in the dispatch table upon determining that the stack reference references a register of a second processor" as claimed in Claims 1 and 31. Therefore, Hamacher fails to cure the deficiencies of Katzman and Morris.

The combined teachings of Katzman, Morris and Hamacher fail to teach or suggest "performing a consistency-preserving operation to recover an in-order value for the instruction referencing the stack from a main memory, and entering the in-order value in the dispatch table upon determining that the stack reference references a register of a second processor" as claimed in claims 1 and 31.

Claims 3, 5, 10 and 11 depend from claim 1. Claims 33, 35 and 36 depend from claim 31. Claims 4 and 34 have been cancelled. The dependent claims are believed to be allowable for at least the reasons given for the independent claims. Reconsideration of the rejection is respectfully requested.

Claims 6-9 and 37-39 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Katzman, in view of Hamacher, Wing (U.S. Patent No. 5,926,832) and Morris. The Examiner

stated essentially that the combined teachings of Katzman, Hamacher, Wing and Morris teach or suggest all the limitations of claims 6-9 and 37-39.

Claims 6-9 depend from claim 1. Claims 37-39 depend from claim 31. The dependent claims are believed to be allowable for at least the reasons given for claims 1 and 31. The Examiner's reconsideration of the rejection is respectfully requested.

Claims 12-19 and 21-30 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Evoy et al. (U.S. Patent No. 5,953,741) and Katzman. The Examiner stated essentially that the combined teachings of Evoy and Katzman teach or suggest all the limitations of claims 12-19 and 21-30.

Claim 12 claims, *inter alia*, "determining whether a load instruction references a location in a local stack using an architecturally defined register for accessing a stack location, wherein each processor comprises a respective local stack" and claim 26 claims, *inter alia*, "determining whether a store instruction references a location in a local stack using an architecturally defined register for accessing a stack location, wherein each processor comprises a respective local stack."

Katzman operates stack registers using "specialized stack operations" (see col. 4, lines 31-35). The specialized stack operations operate on the stack by definition. Thus, Katzman does not teach or suggest, inherently or otherwise, "determining whether a load instruction references a location in a local stack using an architecturally defined register for accessing a stack location", as claimed in Claim 12, or "determining whether a store instruction references a location in a local stack using an architecturally defined register for accessing a stack location", as claimed in Claim 26. Because Katzman's operations are known by definition to reference the stack, there is no determination, inherent or otherwise, of whether the operations reference the stack. Katzman's operations reference only the stack. Therefore, it is not necessary to determine whether the

operations reference a location in a local stack; it is a given, there is no alternative for a stack operation but to reference the stack. Nowhere does Katzman teach or suggest any determination with respect to whether an instruction references the stack, essentially as claimed in Claims 12 and 26. Therefore, Katzman fails to teach or suggest all the limitations of Claims 12 and 26.

Evoy teaches a computer system 10 having a stack 78 utilized for slave processor 50, wherein memory accesses to the stack are accelerated using a stack cache 76 (see Figure 2 and col. 10, lines 42-51). Evoy does not teach or suggest "determining whether a load instruction references a location in a local stack using an architecturally defined register for accessing a stack location", as claimed in Claim 12, or "determining whether a store instruction references a location in a local stack using an architecturally defined register for accessing a stack location", as claimed in Claim 26. Evoy teaches that a DMA unit 72 handles memory accesses to cached portions of the stack that are requested by external devices such as master processor 40 (see col. 10, lines 59-66). Evoy teaches that accesses to the stack cache are handled by the DMA unit 72. Accesses are not made using architecturally defined registers, indeed Evoy does not teach or suggest architecturally defined registers. Thus, no determination is made as to whether an instruction references a location in a local stack using an architecturally defined register, essentially as claimed in Claims 12 and 26. Accordingly, Evoy fails to cure the deficiencies of Katzman.

For at least the foregoing reasons the combined teachings of Katzman and Evoy are believed to fail to teach or suggest, "determining whether a load instruction references a location in a local stack using an architecturally defined register for accessing a stack location", as claimed in Claim 12, or "determining whether a store instruction references a location in a local stack using an architecturally defined register for accessing a stack location", as claimed in Claim 26.

Claims 13-19 and 21-25 depend from claim 12. Claims 27-30 depend from Claim 26. The dependent claims include the elements of their respective independent claims and they are not rendered unpatentable by the cited references for at least the reasons given for the independent claims. At least claims 19 and 28 are believed to be allowable for additional reasons.

Claim 19 claims, "determining, by a first processor, whether the load instruction references a location in any stack, including the local stack, using a register of a second processor, when the load instruction does not reference the location using the architecturally defined register." Claim 28 claims, "determining, by a first processor, whether the store instruction references a location in any stack, including the local stack, using a register of a second processor, when the store instruction does not reference the location using the architecturally defined register."

Katzman teaches a method for managing a system consisting of four top-of-stack registers using a namer register, and a register storing a number of top-of-stack registers containing valid information (SR) and a register containing a top piece of stack information (SM) (see col. 1, line 61 to col. 2, line 6). Katzman does not teach or suggest a plurality of processors, much less determining, by a first processor, whether an instruction references a location in any stack, including the local stack, using a register of a second processor, essentially as claimed in claims 19 and 29. Katzman fails to teach more than one processor. Therefore, Katzman fails to teach or suggest "determining, by a first processor, whether the load instruction references a location in any stack, including the local stack, using a register of a second processor, when the load instruction does not reference the location using the architecturally defined register" as claimed in claim 19 or "determining, by a first processor, whether the store instruction references a location in any stack, including the local stack, using a register of a second processor, when the store instruction does not reference the location using the architecturally defined register" as claimed in claim 28.

Evoy teaches a computer system including a master for processing native code and a slave processor for processing platform-independent program code (see col. 5, lines 36-47). Evoy does not teach or suggest determining, by a first processor, whether an instruction references a location in any stack, including the local stack, using a register of a second processor, essentially as claimed in claims 19 and 29. Evoy teaches that each processor is capable of block the other from the bus or memory (see col. 7, lines 33-51). Evoy does not teach simultaneous use of memory, thus, a situation where an instruction of a first processor references a register of a second processor could not occur. Therefore, Evoy fails to teach or suggest "determining, by a first processor, whether the load instruction references a location in any stack, including the local stack, using a register of a second processor, when the load instruction does not reference the location using the architecturally defined register" as claimed in claim 19 or "determining, by a first processor, whether the store instruction references a location in any stack, including the local stack, using a register of a second processor, when the store instruction does not reference the location using the architecturally defined register" as claimed in claim 28. Accordingly, Evoy fails to cure the deficiencies of Katzman.

Therefore, the combined teachings of Katzman and Evoy fail to teach or suggest "determining, by a first processor, whether the load instruction references a location in any stack, including the local stack, using a register of a second processor, when the load instruction does not reference the location using the architecturally defined register" as claimed in claim 19 or "determining, by a first processor, whether the store instruction references a location in any stack, including the local stack, using a register of a second processor, when the store instruction does not reference the location using the architecturally defined register" as claimed in claim 28.

Reconsideration of the rejection is respectfully requested.

Claim 20 has been rejected under 35 U.S.C. 103(a) as being unpatentble over Evoy in view of Katzman and Wing. The Examiner stated essentially that the combined teachings of Evoy, Katzman and Wing teach or suggest all the limitations of claim 20.

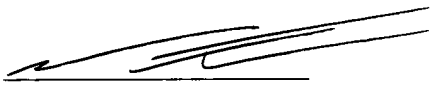
Claim 20 depends from claim 12. Claim 20 is believed to be allowable for at least the reasons given for claim 12. Reconsideration of the rejection is respectfully requested.

Claims 3, 5-7, 9, 33, 35-37 and 39 have been amended to correct various informalities associated with the amendment of claims 1 and 31, including for example, establishing proper antecedent basis.

New claims 40 and 41 depend from claims 1 and 31, respectively. The dependent claims are believed to be allowable for at least the reasons given for claims 1 and 31.

Accordingly, claims 1, 3, 5-31, and 33, 35-41 are believed to be allowable for at least the reasons stated. The Examiner's withdrawal of the rejections is respectfully requested. For the forgoing reasons, the application is believed to be in condition for allowance. Early and favorable reconsideration is respectfully requested.

Respectfully submitted,



Nathaniel T. Wallace
Reg. No. 48,909
Attorney for Applicants

Mailing Address:
F. CHAU & ASSOCIATES, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889